

JOINT 4-D VARIATIONAL STEREO RECONSTRUCTION AND CAMERA CALIBRATION REFINEMENT FOR OCEANIC SEA STATE MEASUREMENTS

Ping-Chang Shih

Guillermo Gallego

Anthony Yezzi

Francesco Fedele

ABSTRACT

Validating modern oceanographic theories using models produced through stereo computer vision principles has recently emerged. Space-time (4-D) models of the ocean surface may be generated by stacking a series of 3-D reconstructions independently generated for each time instant or, in a more robust manner, by simultaneously processing several snapshots coherently in a true “4-D reconstruction.” However, the accuracy of these computer-vision-generated models is subject to the estimations of camera parameters, which may be corrupted under the influence of natural factors such as wind and vibrations. Therefore, removing the unpredictable errors of the camera parameters is necessary for an accurate reconstruction. In this paper, we propose a novel algorithm that can jointly perform a 4-D reconstruction as well as correct the camera parameter errors introduced by external factors. The technique is founded upon variational optimization methods to benefit from their numerous advantages: continuity of the estimated surface in space and time, robustness, and accuracy. The performance of the proposed algorithm is tested using synthetic data produced through computer graphics techniques, based on which the errors of the camera parameters arising from natural factors can be simulated.

INTRODUCTION

Within the past twenty years, computer vision principles have been gradually adopted to create three-dimensional (3-D) models of the ocean surface for measurement and analysis purposes. These computer models are generated through a process known as 3-D reconstruction in the field of stereo computer vision, where the 3-D shape of an object is recovered based on its 2-D projections (e.g. observed images) obtained from various viewpoints.

Three-dimensional reconstruction methods can be categorized into two types: feature-based and variational methods. Due to its simplicity, the former type was adopted by many researchers to reconstruct ocean surfaces [1–7]. Such methods achieve reconstruction in two steps, by first matching corresponding features (e.g. points) across images and then back-projecting them to produce a set of 3-D points (e.g. a point cloud) that represents the spatial positions of the observed 2-D features. However, this type of methods is not particularly effective when applied to a target object that lacks distinctive (e.g. textured) image features, such as the ocean surface, because few object are reconstructed. Consequently, the yielded point cloud representation of the object surface is sparse and therefore inadequate for applications requiring a dense model.

Variational 3-D reconstruction methods have been founded upon the advantages of the calculus of variations to overcome the aforementioned disadvantages of feature-based methods. By converting a computer vision problem into a variational optimization one, these 3-D reconstruction methods approximate the surface of the target object by piece-wise smooth functions [8,9] instead of a collection of 3-D points. Therefore, users can arbitrarily sample the functions to visualize the model at any resolution or to analyze the model at any location. Utilizing this concept, Gallego *et al.* [10–12] proposed a variational framework to reconstruct the space-time model of a patch of ocean surface. In this framework, the reconstructed surface of the object is obtained as the minimizer of a functional that takes into account both the smoothness of the surface and its photometric error. The latter quantifies the discrepancy between the measurements (snapshots of videos) and the “reprojections” of the reconstruction onto measurements via mathematical coordinate transformations and pin-hole projection formulas. The resulting reconstruction consists of two parts: the height or elevation map and the superficial texture pattern (called radiance map) of the ocean surface. When the radiance map is superimposed on top of the elevation map, a computer model of the ocean surface is obtained.

However, the accuracy of all aforementioned computer-vision-generated models strongly depends on the accuracy in the determination of the cameras’ parameters, a technique called camera calibration. These parameters specify the perspective projection operation carried out by the cameras (mapping points in the 3-D world to 2-D sensors) and its inverse operation (back-projection), hence they have a direct effect in the evaluation of reprojection errors. The reconstruction process is very sensitive to the changes of camera parameters, e.g., small deviations of the camera parameters can cause a magnified incorrect reprojection error and, consequently, change the minimizers of the error functional, yielding incorrect reconstruction. Since cameras are installed outdoors in real applications, extrinsic camera parameters—parameters that accounts for the relative orientation and location of the cameras in the scene—are prone to be perturbed by natural factors such as breeze or vibrations. Therefore, the reconstruction of a patch of the ocean surface over a time interval should not be performed alone but be incorporated with a camera calibration refinement technique.

In this paper, we address the problem of reconstructing a patch of the ocean surface over a given period of time in the case that the extrinsic camera parameters might be perturbed by environmental factors. To this end, we adopt a variational optimization framework and jointly estimate the 4-D reconstruction of the ocean surface and the refinement of the camera parameters as the minimizers of an error functional that measures *i)* the photometric error between acquired videos and the reprojection of the 4-D reconstruction onto the videos, *ii)* the spatial and temporal smoothness of the reconstruction, and *iii)* the temporal

variance of the perturbed camera parameters. As a result of iteratively obtaining the minimizers of the error functional, our algorithm can reconstruct the space-time model for the target region and decrease the influence of the errors caused by deviations of the camera parameters on the reconstruction. Since introducing the desired perturbations to the camera parameters and acquiring the ground truth of the reconstruction in real cases are difficult tasks, we validate the algorithm and show its effectiveness by applying it to reconstruct a synthetic ocean surface generated by a computer graphics tool (OpenGL) under the conditions that deviations are deliberately added into the true camera parameters.

METHODOLOGY

Notation and Geometric Image Formation

We use \mathbf{X}^τ to denote the coordinates of a general point on the model observed in the world coordinate system at moment τ . When observed by the i^{th} camera coordinate system, \mathbf{X}^τ can be expressed as $\tilde{\mathbf{X}}_i = (\tilde{X}_i, \tilde{Y}_i, \tilde{Z}_i)^\top$. Coordinates \mathbf{X}^τ and $\tilde{\mathbf{X}}_i$ are linearly related by a rigid body motion, $\tilde{\mathbf{X}}_i = \mathbf{R}_i^\tau \mathbf{X}^\tau + \mathbf{t}_i^\tau$, where \mathbf{R}_i^τ is a three-by-three rotation matrix accounting for the difference between the orientations of the world and the i^{th} camera coordinate systems, and $\mathbf{t}_i^\tau \in \mathbb{R}^3$ is the displacement between the origins of the two systems.

Suppose the apertures of the cameras are small enough and lens are ideal. Thus, the imaging of a 3-D point onto the sensor can be approximated as a pin-hole model without considering the distortion effects of lens; as such, each point $\tilde{\mathbf{X}}_i$ is projected onto a 2-D point \mathbf{x} on the image plane using the pin-hole projection formula: $\mathbf{x} = (\bar{x}, \bar{y})^\top = (\tilde{X}_i/\tilde{Z}_i, \tilde{Y}_i/\tilde{Z}_i)^\top$.

Up to now, \mathbf{x} is given with respect to the i^{th} camera coordinate system, so one more transformation is required to convert it to pixel coordinates $\hat{\mathbf{x}} = (\hat{x}, \hat{y})^\top$, i.e., the position of \mathbf{x} on a CCD image plane. Such a transformation is given by

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} L_x^i & 0 \\ 0 & L_y^i \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} + \begin{bmatrix} x_0^i \\ y_0^i \end{bmatrix},$$

where the internal parameters of the camera (L_x^i and L_y^i are the focal lengths and $(x_0^i, y_0^i)^\top$ is the principal point of the i^{th} camera, in pixel units) are compactly represented by the intrinsic camera matrix [13]

$$\mathbf{K}_i = \begin{bmatrix} L_x^i & 0 & x_0^i \\ 0 & L_y^i & y_0^i \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that we use bold fonts to represent vector quantities and add superscript τ to the extrinsic parameters of the cameras— \mathbf{R}_i^τ and \mathbf{t}_i^τ —to emphasize their temporal dependence due to

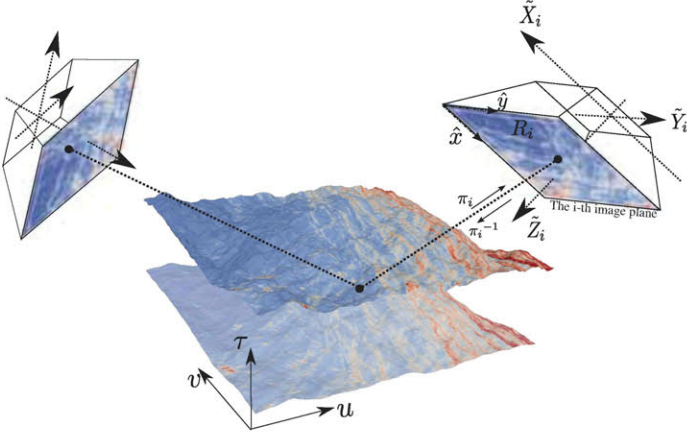


FIGURE 1: Illustration of the space-time reconstruction of a patch of ocean surface. The synchronized cameras film the same area for a period of time while the extrinsic camera parameters may be perturbed by environmental factors. The stacked graphs in the middle symbolize the ocean surface at different moments in time. Each point on the space-time reconstruction is represented as \mathbf{X}^τ , which is imaged by the cameras and then converted to pixel coordinates $\hat{\mathbf{x}}$.

the possible influence of environmental factors, whereas such a notation is not used in case of the intrinsic camera parameters in \mathbf{K}_i since we assume that they remain constant or that the effect on the system of their small variations is negligible compared to that of the extrinsic parameters. In addition, to simplify notation, τ is not particularly used on some symbols such as $\hat{\mathbf{x}}$, \mathbf{x} , and $\hat{\mathbf{X}}_i$, even though they are time dependent. All symbols and their relationships are geometrically shown in Fig. 1.

Proposed Error Functional

To jointly reconstruct a dense space-time model of a patch of ocean surface and simultaneously refine the extrinsic camera parameters, we propose the following error functional

$$E(f, Z, \boldsymbol{\lambda}) = E_{\text{data}}(f, Z, \boldsymbol{\lambda}) + E_{\text{geom}}(Z) + E_{\text{rad}}(f) + E_{\text{cam}}(\boldsymbol{\lambda}). \quad (1)$$

In this functional, Z and f are functions of three variables (u, v, τ) with domain in $U_T := U \times [0, T]$, where $\mathbf{u} = (u, v)^\top \in U$ are spatial variables and $\tau \in [0, T]$ is the temporal variable (as illustrated in Fig. 1). From the visual perspective, $Z(u, v, \tau)$ and $f(u, v, \tau)$ represent the shape and the texture of the model on parameter point (u, v) at time τ (as shown in Fig. 2), respectively, so $\mathbf{X}^\tau = (u, v, Z(u, v, \tau))^\top$. The vector function $\boldsymbol{\lambda}(\tau) = (\lambda^1(\tau), \dots, \lambda^m(\tau))^\top$ denotes the extrinsic camera parameters estimated within the observation interval of duration T . The func-

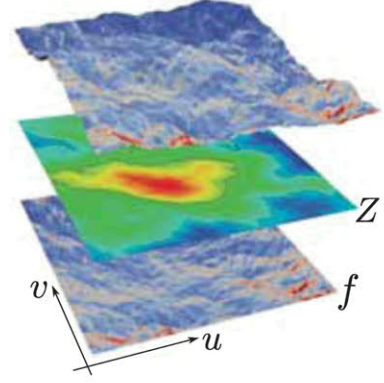


FIGURE 2: Relationship between the elevation map (Z), the radiance map (f), and the reconstructed surface for a given time τ . The reconstructed surface is formed when each spatial coordinate $Z(u, v, \tau)$ is tinted with “color” or intensity value $f(u, v, \tau)$.

tion λ^j represents a single camera parameter. The minimizers of this functional—a special set of $(f, Z, \boldsymbol{\lambda})$ —are expected to optimally fit the visual measurements and appropriately approximate the superficial properties on the observed ocean surface and the temporal variations of the camera parameters. Therefore, E_{data} , E_{geom} , E_{rad} , and E_{cam} are designed as follows:

$$E_{\text{data}} = \int_T \sum_{i=1}^{N_c} \int_{R_i} \frac{1}{2} (I_i(\hat{\mathbf{x}}) - f(\pi_i^{-1}(\hat{\mathbf{x}})))^2 d\hat{\mathbf{x}} d\tau, \quad (2)$$

$$E_{\text{geom}} = \frac{\alpha}{2} \int_T \int_U (Z_u^2 + Z_v^2 + Z_\tau^2) d\mathbf{u} d\tau, \quad (3)$$

$$E_{\text{rad}} = \frac{\beta}{2} \int_T \int_U (f_u^2 + f_v^2 + f_\tau^2) d\mathbf{u} d\tau, \quad (4)$$

$$E_{\text{cam}} = \frac{\gamma}{2} \int_T \int_{\tau-w}^{\tau+w} \frac{\|\boldsymbol{\lambda}(p) - \boldsymbol{\mu}(\tau; w)\|^2}{2w} dp d\tau, \quad (5)$$

where $\alpha, \beta, \gamma > 0$. In the data fidelity term ((2)), N_c is the number of cameras, R_i is the reprojection region of the reconstruction at time τ on the i^{th} image (denoted by I_i), and $I_i(\hat{\mathbf{x}})$ is the image intensity at pixel $\hat{\mathbf{x}}$. The forward perspective projection (the aforementioned coordinate transformations from \mathbf{X}^τ to $\hat{\mathbf{x}}$ in the previous section) is represented by π_i , and, by abuse of notation, its inverse operation is represented by π_i^{-1} . In the formulation, the first term of E_{data} ((2)) quantifies the sum of the discrepancies between the measurements (the acquired synchronized stereo videos) and the reprojections of the reconstructed surface onto the image planes.

Because the target object is the ocean surface, the shape (Z) and the texture (f) can be assumed to be smooth functions with respect to time and space, which is formulated in (3) and (4). The term E_{geom} measures the spatial and temporal smoothness of Z through its derivatives Z_u , Z_v , and Z_τ , whereas E_{rad} does the

same for f .

Given that natural factors such as breeze or vibrations smoothly influence the extrinsic camera parameters, we design E_{cam} to restrict the temporal behavior of λ . Instead of penalizing the temporal changes of λ in terms of derivatives, as it is done with Z and f in (3) and (4), E_{cam} is designed to penalize the temporal variations of λ with respect to a local variance, as expressed by (5). In the equation, $\mu(\tau; w)$ represents the local mean of λ in an interval of duration $2w$ centered around time τ , that is,

$$\mu(\tau; w) = \frac{1}{2w} \int_{\tau-w}^{\tau+w} \lambda(q) dq. \quad (6)$$

By choosing an appropriate w , we can eliminate different types of perturbations on the camera parameters that are otherwise not possible to be removed using penalties on the derivatives.

Minimizers of the Error Functional The minimizers of (1) are determined by the zero functional derivatives with respect to the arguments (denoted by $\frac{\delta E}{\delta Z} = 0$, $\frac{\delta E}{\delta f} = 0$, and $\frac{\delta E}{\delta \lambda} = \mathbf{0}$, i.e., $\frac{\delta E}{\delta \lambda^j} = 0$ for all j). Methods for deriving the analytical forms of $\frac{\delta E}{\delta Z}$ and $\frac{\delta E}{\delta f}$ are explored in [12]. Although the error functional in [12] differs from (1), $\frac{\delta E}{\delta Z} = 0$ and $\frac{\delta E}{\delta f} = 0$ yield the same set of partial differential equations (PDEs),

$$g(Z, f) - \alpha \Delta Z = 0 \text{ in } U_T, \quad (7)$$

$$-\sum_{i=1}^{N_c} (I_i - f) J_i - \beta \Delta f = 0 \text{ in } U_T, \quad (8)$$

$$b(Z, f) + \alpha \frac{\partial Z}{\partial \mathbf{v}} = 0 \text{ on } \partial U_T, \quad (9)$$

$$\beta \frac{\partial f}{\partial \mathbf{v}} = 0 \text{ on } \partial U_T. \quad (10)$$

In the equations above, $g(Z, f)$ and $b(Z, f)$ are nonlinear terms due to the data fidelity component of the error functional,

$$g(Z, f) = \nabla_{(u,v)} f \cdot \sum_{i=1}^{N_c} |K_i| \tilde{Z}_i^{-3} (I_i - f) (u - C_i^1, v - C_i^2),$$

$$b(Z, f) = \sum_{i=1}^{N_c} \frac{1}{2} (I_i - f)^2 |K_i| \tilde{Z}_i^{-3} ((u - C_i^1) \mathbf{v}^u + (v - C_i^2) \mathbf{v}^v)$$

and J_i is the Jacobian of the change of variables from U to the i^{th} image plane, i.e., $J_i = |K_i| \tilde{Z}_i^{-3} \max(0, -\langle \mathbf{X}^\tau - C_i, \mathbf{X}_u^\tau \times \mathbf{X}_v^\tau \rangle)$ according to the derivations in [11]. The symbol ∂ in (9) and (10) has two different and standard meanings: it represents the directional derivative operator, as in $\frac{\partial Z}{\partial \mathbf{v}}$, or indicates the boundary of the domain of Z and f , as in ∂U_T . Finally, $\mathbf{v} = (v^u, v^v, v^\tau)^\top$ is

the unit outward normal on ∂U_T , and $C_i = (C_i^1, C_i^2, C_i^3)^\top$ is a vector with the world coordinates of the i^{th} camera center.

To compute $\frac{\delta E}{\delta \lambda}$, we augment λ with an artificial time variable t and differentiate λ with respect to t using chain rule relationship, yielding $\frac{\partial E}{\partial t} = \left\langle \frac{\delta E}{\delta \lambda^j}, \lambda_t^j \right\rangle_{L^2(T)}$, where $\langle \cdot, \cdot \rangle_{L^2(T)}$ is the L^2 -inner product operator between functions defined over the interval of duration T . This lays down the path toward obtaining an extrema of E by evolving λ with respect to t using a descent method.

Thus, differentiating E with respect to t leads to

$$\frac{\partial E}{\partial t} = \frac{\partial E_{\text{data}}}{\partial t} + \frac{\partial E_{\text{cam}}}{\partial t}, \quad (11)$$

where the smoothness terms in E_{geom} and E_{rad} vanish because they do not depend on the camera parameters λ . Next, let us show how to compute each term in (11), starting with the data fidelity one: Since R_i^τ and \mathbf{t}_i^τ are the parameters of the i^{th} camera, we can drop the summation symbol in (2) when differentiating E with respect to t . Hence,

$$\frac{\partial E_{\text{data}}}{\partial t} = \int_T \frac{\partial}{\partial t} \left(\int_{R_i} \frac{1}{2} (I_i(\hat{\mathbf{x}}) - f(\pi_i^{-1}(\hat{\mathbf{x}})))^2 d\hat{\mathbf{x}} \right) d\tau. \quad (12)$$

Operator $\frac{\partial}{\partial t}$ in (12) cannot be directly moved inside \int_{R_i} because R_i depends on λ^j , and consequently on t . In this case, deriving the analytical form of (12) requires the use of Reynolds' transport theorem and a change of variables, which was completed by Shih *et al.* in [14]. Compared with the assumption in this paper, the extrinsic camera parameters discussed in [14] are regarded as temporally constant variables, which, nevertheless, does not affect the reasoning process of the derivations in this case. Hence, the right hand side of (12) is ultimately differentiated as

$$\begin{aligned} \frac{\partial E_{\text{data}}}{\partial t} = & \left\langle \lambda_t^j, \int_{\partial U} \frac{[I_i(\pi_i(\mathbf{X}^\tau)) - f(\mathbf{X}^\tau)]^2}{2} \left\langle \frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial \lambda^j}, Q \frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial s} \right\rangle ds \right. \\ & \left. + \int_U [f(\mathbf{X}^\tau) - I_i(\pi_i(\mathbf{X}^\tau))] \left\langle \frac{\partial \mathbf{X}^\tau}{\partial \lambda^j}, \nabla_s f(\mathbf{X}^\tau) \right\rangle J_i du \right\rangle_{L^2(T)}, \end{aligned} \quad (13)$$

where $\langle \cdot, \cdot \rangle$ is the usual inner product operator in \mathbb{R}^n ($\langle a, b \rangle = a^\top b$) and Q is a planar (two-by-two) rotation matrix that rotates a unit tangent vector to the corresponding outward normal. In addition, $\nabla_s f$ stands for $(\frac{\partial f}{\partial \mathbf{X}^\tau})^\top$, which is $(f_u, f_v, 0)^\top$.

With different types of camera parameters, $\frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial \lambda^j}$ and $\frac{\partial \mathbf{X}^\tau}{\partial \lambda^j}$ in (13) possess different analytical forms. For example, if

¹Although $C_i = -(\mathbf{R}_i^\tau)^\top \mathbf{t}_i^\tau$ depends on τ , the corresponding superscript is not explicitly shown.

$\boldsymbol{\lambda} = \mathbf{t}_i^\tau$, according to [14],

$$\frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} L_x^i \tilde{\mathbf{Z}}_i^{-1} & 0 & -L_x^i \tilde{\mathbf{X}}_i \tilde{\mathbf{Z}}_i^{-2} \\ 0 & L_y^i \tilde{\mathbf{Z}}_i^{-1} & -L_y^i \tilde{\mathbf{Y}}_i \tilde{\mathbf{Z}}_i^{-2} \end{bmatrix}$$

and

$$\frac{\partial \mathbf{X}^\tau}{\partial \boldsymbol{\lambda}} = -(\mathbf{R}_i^\tau)^{-1} \left(\mathbf{Id}_{3 \times 3} - \frac{\tilde{\mathbf{X}}_i \mathbf{N}_i^\top}{\tilde{\mathbf{X}}_i^\top \mathbf{N}_i} \right),$$

where \mathbf{N}_i is an outward normal on the reconstructed model observed in the i^{th} camera frame. Furthermore, [14] indicates that if $\boldsymbol{\lambda} = \tilde{\omega}_i^\tau$ ($\tilde{\omega}_i^\tau$ is the rotation vector of \mathbf{R}_i^τ in exponential coordinates), then

$$\frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial \boldsymbol{\lambda}} = - \begin{bmatrix} L_x^i \tilde{\mathbf{Z}}_i^{-1} & 0 & -L_x^i \tilde{\mathbf{X}}_i \tilde{\mathbf{Z}}_i^{-2} \\ 0 & L_y^i \tilde{\mathbf{Z}}_i^{-1} & -L_y^i \tilde{\mathbf{Y}}_i \tilde{\mathbf{Z}}_i^{-2} \end{bmatrix} \mathbf{R}_i^\tau [\mathbf{X}^\tau]_\times$$

and²

$$\frac{\partial \mathbf{X}^\tau}{\partial \boldsymbol{\lambda}} = (\mathbf{R}_i^\tau)^{-1} \left(\mathbf{Id}_{3 \times 3} - \frac{\tilde{\mathbf{X}}_i \mathbf{N}_i^\top}{\tilde{\mathbf{X}}_i^\top \mathbf{N}_i} \right) \mathbf{R}_i^\tau [\mathbf{X}^\tau]_\times.$$

Next, let us deal with the second term in (11). Taking into account that $\boldsymbol{\lambda} \equiv \boldsymbol{\lambda}(p, t)$ depends on t , and therefore $\boldsymbol{\mu} \equiv \boldsymbol{\mu}(\tau, t; w)$ and $E_{\text{cam}}(\boldsymbol{\lambda})$ depend on t , too, the second term in (11) is computed as follows:

$$\frac{\partial E_{\text{cam}}}{\partial t} = \frac{\gamma}{2w} \int_T \int_{\tau-w}^{\tau+w} (\boldsymbol{\lambda} - \boldsymbol{\mu})^\top \frac{\partial}{\partial t} (\boldsymbol{\lambda} - \boldsymbol{\mu}) dp d\tau.$$

Because $\boldsymbol{\mu}_t$ does not depend on p , $\boldsymbol{\mu}_t$ can be moved out of the inner integral. As such,

$$\int_{\tau-w}^{\tau+w} (\boldsymbol{\lambda} - \boldsymbol{\mu})^\top \boldsymbol{\mu}_t dp = \underbrace{\left(\int_{\tau-w}^{\tau+w} (\boldsymbol{\lambda} - \boldsymbol{\mu}) dp \right)^\top}_{=0} \boldsymbol{\mu}_t = 0,$$

so we are left with

$$\frac{\partial E_{\text{cam}}}{\partial t} = \frac{\gamma}{2w} \int_T \int_{\tau-w}^{\tau+w} (\boldsymbol{\lambda} - \boldsymbol{\mu})^\top \boldsymbol{\lambda}_t dp d\tau. \quad (14)$$

To further simplify (14), let us swap the order of integration by using a window function $W(\tau, p; w) = H(p - \tau + w) - H(p - \tau - w)$. H is the Heaviside step function defined as

²Matrix $[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$ is the cross product matrix associated to vector $\mathbf{a} = (a_1, a_2, a_3)^\top \in \mathbb{R}^3$ so that $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b}$ for all $\mathbf{b} \in \mathbb{R}^3$.

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases},$$

which accounts for the fact that the integrand of (14) may not vanish if $\tau - w \leq p \leq \tau + w$, or equivalently, if $p - w \leq \tau \leq p + w$. Then, recalling that $\boldsymbol{\lambda} \equiv \boldsymbol{\lambda}(p, t)$ and $\boldsymbol{\mu} \equiv \boldsymbol{\mu}(\tau, t; w)$ in (14), it becomes

$$\begin{aligned} \frac{\partial E_{\text{cam}}}{\partial t} &= \frac{\gamma}{2w} \int_T \int_T \boldsymbol{\lambda}_t^\top (\boldsymbol{\lambda} - \boldsymbol{\mu}) W(\tau, p; w) d\tau dp \\ &= \int_T \boldsymbol{\lambda}_t^\top \left(\frac{\gamma}{2w} \int_T (\boldsymbol{\lambda} - \boldsymbol{\mu}) W(\tau, p; w) d\tau \right) dp \\ &= \int_T \boldsymbol{\lambda}_t^\top \left(\frac{\gamma}{2w} \int_{p-w}^{p+w} (\boldsymbol{\lambda} - \boldsymbol{\mu}) d\tau \right) dp \\ &= \left\langle \boldsymbol{\lambda}_t, \frac{\delta E_{\text{cam}}}{\delta \boldsymbol{\lambda}} \right\rangle_{L^2(T)}, \end{aligned}$$

using the vectorial version of $\langle \cdot, \cdot \rangle_{L^2(T)}$. Therefore, we identify

$$\begin{aligned} \frac{\delta E_{\text{cam}}}{\delta \boldsymbol{\lambda}}(p) &= \frac{\gamma}{2w} \int_{p-w}^{p+w} (\boldsymbol{\lambda} - \boldsymbol{\mu}) d\tau \\ &= \gamma \left(\boldsymbol{\lambda}(p) - \frac{1}{2w} \int_{p-w}^{p+w} \left[\frac{1}{2w} \int_{\tau-w}^{\tau+w} \boldsymbol{\lambda}(q) dq \right] d\tau \right), \end{aligned} \quad (15)$$

where in the last line we substituted the definition (6) for the local mean $\boldsymbol{\mu}$ and omitted the dependence of $\boldsymbol{\lambda}$ with respect to t .

Combining the functional gradients (with respect to the camera parameters) of the data fidelity and the camera refinement components of the error functional, the gradient descent flow corresponding to $\frac{\delta E}{\delta \boldsymbol{\lambda}} = \mathbf{0}$ is $\boldsymbol{\lambda}_t = -\frac{\delta E}{\delta \boldsymbol{\lambda}}$. Therefore,

$$\begin{aligned} \lambda_t^j(\tau) &= - \left\{ \int_{\partial U} \frac{[I_i(\pi_i(\mathbf{X}^\tau)) - f(\mathbf{X}^\tau)]^2}{2} \left\langle \frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial \lambda^j}, \mathcal{Q} \frac{\partial(\pi_i(\mathbf{X}^\tau))}{\partial s} \right\rangle ds \right. \\ &\quad + \int_U [f(\mathbf{X}^\tau) - I_i(\pi_i(\mathbf{X}^\tau))] \left\langle \frac{\partial \mathbf{X}^\tau}{\partial \lambda^j}, \nabla_{\mathbb{S}} f(\mathbf{X}^\tau) \right\rangle J_i du \\ &\quad \left. + \gamma \lambda^j(\tau) - \frac{\gamma}{4w^2} \int_{\tau-w}^{\tau+w} \left(\int_{z-w}^{z+w} \lambda^j(q) dq \right) dz \right\}, \quad \forall j. \end{aligned} \quad (16)$$

EXPERIMENTS

To validate the proposed algorithm, we conduct experiments on synthetic data, which are adopted because the true extrinsic camera parameters of a stereo computer vision apparatus and the space-time model of a real ocean surface are difficult to obtain. Therefore, we will rely on a computer graphics library—OpenGL—to generate a synthetic ocean surface for the experiments [15], which is based on the work by Tessendorf [16]. Although the synthetic model generated through OpenGL may not fully approximate the physical properties of real data, OpenGL does offer users the flexibility to quantify and control various types of camera parameters. Based on this property, we can simulate the situations in which the extrinsic camera parameters are perturbed by external factors.

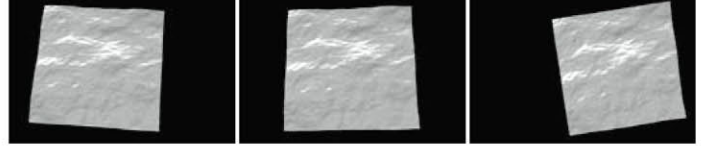
Synthetic Data

Computer graphics and 3-D reconstruction applications are reverse operations of each other. In computer graphics, imaginary objects are created in a virtual world and can be viewed from various angles. Using computer graphics tools, such as OpenGL, we can create a region of deforming ocean waves in a virtual world and set up multiple virtual cameras to acquire videos of the synthetic ocean surface. The synthetic ocean waves will be a dense 3-D point cloud deforming with respect to time in which the coordinates (in the virtual world) of all points are known (because we create them), and textures will be rendered on the point cloud as the superficial pattern of the synthetic waves. The videos—the contents shown in the application programming interface (API) windows—can be read from the memory of the computer graphics card. However, because the reference frame of the virtual world is different from that of our algorithm, we must perform a coordinate transformation (a rigid body motion) to express the simulated extrinsic camera parameters in the reference system used by the algorithm. After this information is ready, we can add errors to the camera parameters, reconstruct the model of the synthetic waves, and apply our calibration refinement algorithm to see how the reconstruction is rectified. By deliberately discretizing the elevation and radiance maps of the reconstructed model with the same number of samples as the input point cloud, we can verify the accuracy of the reconstructed model by point-wise comparison with respect to the input point cloud and radiance.

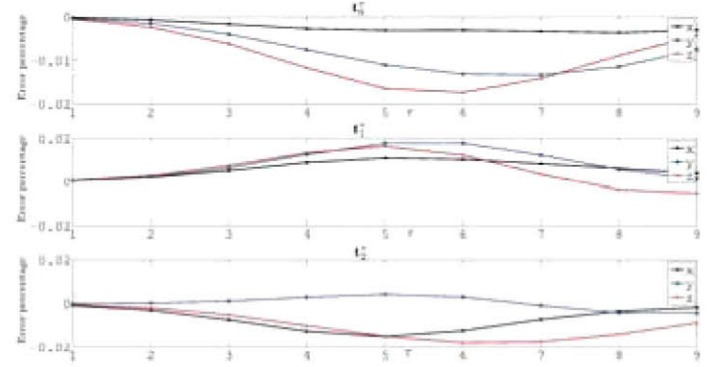
The configuration of the synthetic data is illustrated in Fig. 3. A deforming ocean surface is created in a virtual world and observed from three different positions (as shown in Fig. 3a) for a specified time interval. Artificial errors are added to the true extrinsic camera parameters converted from the configuration of virtual scene developed using OpenGL. Since the purpose of this experiment is to explore the performance of our algorithm under the smooth influence of environmental factors, the artificial errors are deliberately smoothed within the observation interval (nine temporal samples), as shown in Figs. 3b and 3c.

We particularly take out the inputs and outputs at temporal samples 1, 3, 5, 7, and 9 for demonstration. The visual observations of the deforming ocean surface are listed in Fig. 4a, and the elevation maps produced through OpenGL at those time samples are listed in Fig. 4b. The width and height of the surface are set to be 7,475.2 mm, and the range of the height is configured to be $[-500, 450]$ mm, as marked by the hue bars.

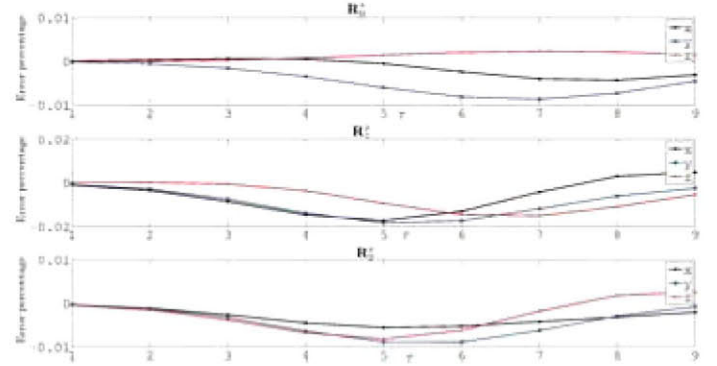
We initially set $\alpha = 5$, $\beta = 50$, $\gamma = 500$, and $w = 3$ for numerically solving (7) – (10) and (16). These parameters produce a rough reconstruction. Afterward, we continue to solve the aforementioned equations while α , β , γ , and w are gradually reduced to $\alpha = 0.2$, $\beta = 0.025$, $\gamma = 100$, and $w = 1$. Eventually, the elevation maps generated through this process are shown in Fig. 4c. In addition, the differences between Figs. 4b and 4c are shown in Fig. 4d. All hue bars are given in millimeter units.



(a) A patch of synthetic ocean surface observed from three different viewpoints at a particular time.



(b) Errors introduced to the translational part \mathbf{t}_i^T of the extrinsic parameters.

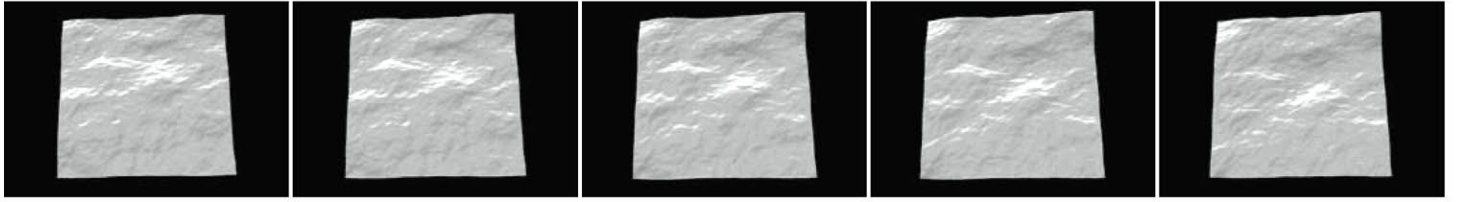


(c) Errors introduced to the rotational part \mathbf{R}_i^T of the extrinsic parameters.

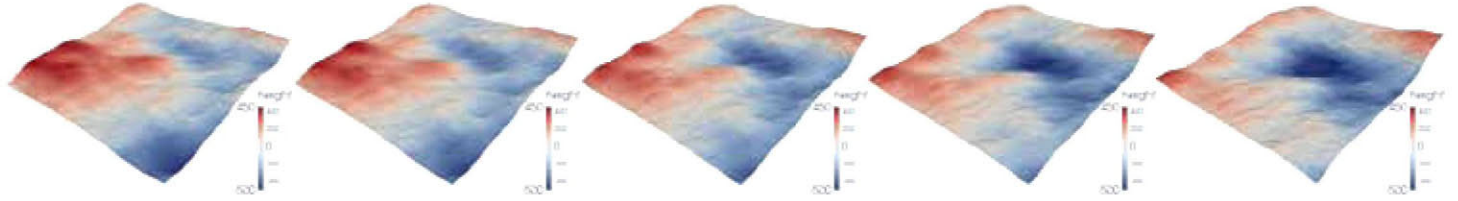
FIGURE 3: Configuration of the synthetic data: Artificial errors, temporally smooth, are added to corrupt the true camera parameters converted from OpenGL to simulate the effects imposed by natural factors on a 3-D reconstruction camera system. Note that the vertical axes indicate that errors are limited to be less than 2% of the true values.

An experiment where the calibration refinement is not performed is conducted and shown in Fig. 4e for comparison. In this experiment, only (7)–(10) are solved; (16) is not used (i.e., $\gamma = 0$). The error maps between the outcomes of this experiment and the inputs are shown in Fig. 4f.

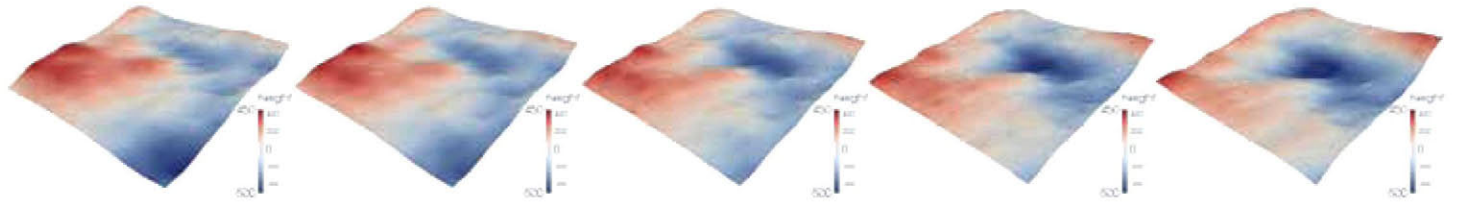
Because the artificial errors introduced at time 1 are small, the corresponding error maps (the first images in Fig. 4d and 4f) exhibit small differences with respect to the ground truth shown in Fig. 4b. With increasing errors added to the subsequent time steps, the error maps display more and more significant differ-



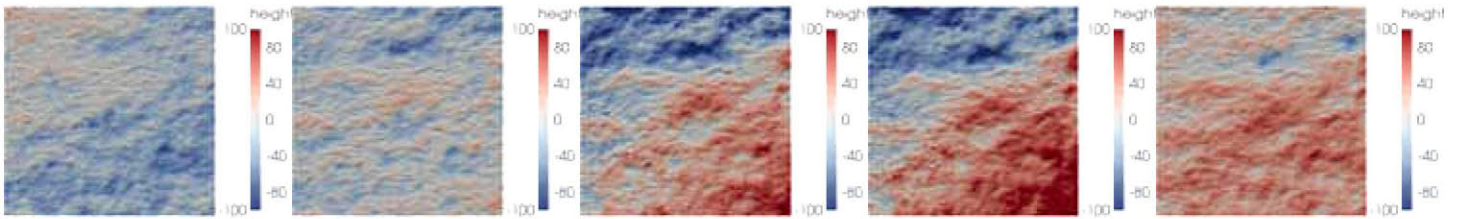
(a) Snapshots taken at time samples 1, 3, 5, 7, and 9 by the middle camera.



(b) Elevation maps generated through OpenGL at time samples 1, 3, 5, 7, and 9.



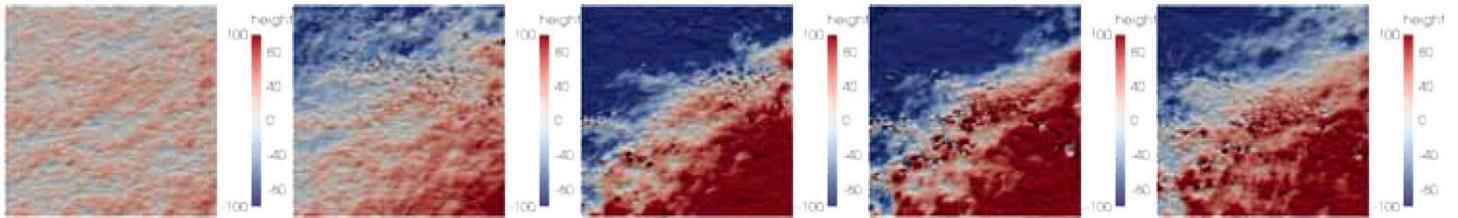
(c) Elevation maps (at time samples 1, 3, 5, 7, and 9) extracted from the space-time reconstruction generated through the algorithm proposed in this paper.



(d) Errors between Figs. 4b and 4c.



(e) Elevation maps (at time samples 1, 3, 5, 7, and 9) extracted from the space-time reconstruction without calibration refinement.



(f) Errors between Figs. 4b and 4e.

FIGURE 4: Input and output of the space-time reconstruction process, with (middle) and without (bottom) camera calibration refinement.

ences. In addition, the effect of the calibration refinement can be observed from the comparison between Figs. 4d and 4f.

In the case in which the reconstruction and the calibration refinement are jointly performed (as shown in Fig. 4d), most of the points in the error maps at time samples 1, 3, 5, 7, and 9 fall within the ± 40 mm range. By contrast, when the reconstruction is executed without calibration refinement, a very large portion of the error maps exhibit colors beyond the range of the hue bars.

Note that even *i*) with relatively large errors added to the camera parameters and *ii*) without the calibration refinement, the reconstruction at time samples 5, 7, and 9 in Fig. 4e still roughly resemble the ground truth in Fig. 4b, which is mainly due to the temporal coherence imposed on the space-time reconstruction, as shown in (3) and (4).

Conclusion

In the context of stereo vision systems to measure ocean waves in space and time, we addressed the recovery of the surface of the ocean and the refinement of the extrinsic camera parameters as a joint problem. To this end, we developed an algorithm based on the minimization of an error functional that incorporates a penalty to filter the variations of the camera parameters. The simulations carried out demonstrated the improvement in the surface reconstruction step of the proposed technique for a specific example. In future work, we intend to carry out a more thorough evaluation of the deviations of the camera parameters that this technique can tolerate.

The work presented is another step toward accurate stereoscopic means of determining the 4-D ocean surface features from video recordings, which ultimately offers the possibility of overcoming many uncertainties related to conventional wave measuring devices. The topic we addressed, reducing the errors associated with camera deviations, is particularly important in ocean engineering since such measurement systems would find application on offshore and gas facilities in which significant camera motions would be unavoidable.

ACKNOWLEDGMENT

OceanFFT project, CUDA SDK. This work has been partially supported by the Office of Naval Research Grant BAA 09-012: "Ocean Wave Dissipation and Energy Balance (WAVE-DB): toward reliable spectra and first breaking statistics" and by the Ministerio de Economía y Competitividad of the Spanish Government under project TEC2010-20412 (Enhanced 3DTV).

REFERENCES

- [1] Santel, F., Heipke, C., Konnecke, S., and Wegmann, H., 2002. "Image sequence matching for the determination of three-dimensional wave surfaces". *International archives of photogrammetry remote sensing and spatial information sciences*, **34**(5), pp. 596–600.
- [2] Santel, F., Linder, W., and Heipke, C., 2004. "Stereoscopic 3D-image sequence analysis of sea surfaces". In Proc. ISPRS Commission V Symposium, Vol. 35, pp. 708–712.
- [3] Benetazzo, A., 2006. "Measurements of short water waves using stereo matched image sequences". *Coastal engineering*, **53**(12), pp. 1013–1032.
- [4] Wanek, J., and Wu, C., 2006. "Automated trinocular stereo imaging system for three-dimensional surface wave measurements". *Ocean engineering*, **33**(5), pp. 723–747.
- [5] MacHutchon, K., and Liu, P., 2007. "Measurement and analysis of ocean wave fields in four dimensions". In OMAE, Vol. 1, pp. 923–927.
- [6] de Vries, S., et al., 2011. "Remote sensing of surf zone waves using stereo imaging". *Coastal Engineering*, **58**(3), pp. 239–250.
- [7] Bechle, A. J., and Wu, C. H., 2011. "Virtual wave gauges based upon stereo imaging for measuring surface wave characteristics". *Coastal Engineering*, **58**(4), pp. 305–316.
- [8] Faugeras, O., and Keriven, R., 1998. "Variational principles, surface evolution, pde's, level set methods and the stereo problem". *IEEE Trans. Image Processing*, **7**(3), pp. 336–344.
- [9] Yezzi, A., and Soatto, S., 2003. "Stereoscopic segmentation". *Int. Journal of Computer Vision*, **53**(1), pp. 31–43.
- [10] Gallego, G., Yezzi, A., Fedele, F., and Benetazzo, A., 2011. "A Variational Stereo Method for the Three-Dimensional Reconstruction of Ocean Waves". *IEEE Trans. Geoscience and Remote Sensing*, **49**(11), pp. 4445–4457.
- [11] Gallego, G., 2011. "Variational image processing algorithms for the stereoscopic space-time reconstruction of water waves". PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA.
- [12] Gallego, G., Yezzi, A., Fedele, F., and Benetazzo, A., 2012. "Space-time Reconstruction of Oceanic Sea States via Variational Stereo Methods". In Proc. ISOPE 22nd Int. Conf. on Offshore and Polar Engineering, Vol. 3, pp. 732–739.
- [13] Hartley, R., and Zisserman, A., 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- [14] Shih, P.-C., Gallego, G., Yezzi, A., and Fedele, F., 2013. "Improving 3-D Variational Stereo Reconstruction of Oceanic Sea States by Camera Calibration Refinement". In Proc. ASME 32nd Int. Conf. on Ocean, Offshore and Arctic Engineering.
- [15] NVIDIA Corporation. NVIDIA CUDA SDK - Physically-Based Simulation. http://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86_website/Physically-Based_Simulation.html.
- [16] Tessendorf, J., 1999-2004. "Simulating Ocean Water". In SIGGRAPH course notes. <http://jerrytessendorf.blogspot.com.es/>.